Table of Contents

S.No	Date	Title	Page No	Signature
8		**Modelling UML Use Case Diagrams and Capturing Use Case Scenarios** 8.1.1 Use Case Description 8.1.2 Use Case Diagram		

teamgantt
Created with Free Edition

Integrated Music Tools	start	end	81h	97%
Problem Statements	**10/03/22**	**17/03/22**	**9h**	**100%**
Identify the Software Project Problem...	10/03	11/03	3	100%
Create Business Case	10/03	17/03	6	100%
Stakeholders and Process Models	**17/03/22**	**24/03/22**	**10h**	**100%**
Identifying Stakeholders	17/03	18/03	4	100%
User story	18/03	19/03	0	100%
Identify the Appropriate Process Mod...	19/03	20/03	2	100%
Arrive at a Problem Statement	20/03	21/03	0	100%
Perspective Model	21/03	22/03	3	100%
Comparison between Agile and Wate...	22/03	24/03	1	100%
Identifying The Requirements From...	**24/03/22**	**30/03/22**	**11h**	**100%**
Requirements	24/03	25/03	3	100%
System Requirements	24/03	27/03	2	100%
Functional Requirements	26/03	29/03	5	100%
Non-Functional Requirements	29/03	30/03	1	100%
Project Plan and Project Effort Bas...	**30/03/22**	**31/03/22**	**9h**	**100%**
Project Plan	30/03	31/03	6	100%
Identify Job Roles and Responsibilities	31/03	31/03	3	100%
Project Effort Based On Resources	**01/04/22**	**03/04/22**	**9h**	**100%**
Work Breakdown Structure	01/04	02/04	4	100%
Risk Analysis	02/04	03/04	5	100%
Estimation of Project Metrics	**03/04/22**	**05/04/22**	**6h**	**100%**
Function Point Analysis	03/04	05/04	2	100%
COCOMO Model	04/04	05/04	4	100%
Design	**05/04/22**	**07/04/22**	**8h**	**100%**
System Design	05/04	07/04	8	100%
Modelling UML Use Case Diagrams...	**08/04/22**	**10/04/22**	**10h**	**100%**
Use case description	08/04	08/04	3	100%
Use case diagram	09/04	10/04	7	100%
E-R modeling from the Problem Sta...	**10/04/22**	**11/04/22**	**9h**	**47%**
ER Diagram	11/04	11/04	4	2%
ER Diagram Description	10/04	11/04	5	70%

EXP NO: 1

DATE:

1.1 Problem Statements:

1.1.1 The Problem Statement:

Introduction:

Music is an integral part of all our lives especially during the lockdown to deal with a lot of emotions. There are a lot of advancement in the music platforms but each of them comes with their own disadvantages. The main aim is to discard certain discomforts for listeners. The software which is to be designed delivers tools to enhance the musical experience using intense machine learning algorithms. The platform further intends to connect various music enthusiast with similar taste thereby opening a new world.

Problems:

1. Unable to share music to listeners across platforms:

Each platform has their own set of users, but the music has no bounds. There is no facilities to share them across platforms.

2. Absence of media platform to share music and meet people based on taste:

Music binds us all. There are no facilities to meet people are share our songs. The software intends to provide a space to explore many listeners' playlist and share their favorite songs.

3. There is no personalized mood for each day:

There are premixed playlists but not a dynamic one to suggest us based on the current mood.

4. There are no means to explore music based on our taste

Proposal:

The software designed offers tools to integrate music across platforms and makes sure user has the best music experience. The software moreover helps listeners to connect and explore music based on their taste and mood of the hour.

Platforms used:

1. Spotify
2. Apple music
3. YouTube

1.1.2 One Page Business Case Template

Date	24/03/2022
Submitted By	Ashwin Ram RA2011026020027 Krishnan Narayanan RA2011026020030 Anuradha Arumugam RA2011026020043
Project Title	Integrated Music Tools

The Project: In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- It is a set of tools that helps user to enjoy music more.
- The tools include features such as
  - creating music playlists across platforms
  - songs suggestion based on mood
  - exploration of music
  - meeting new people with similar taste in music.

The History: In bullet points, describe the current situation.

- The software was designed to overcome barriers in sharing music.
- Moreover, music binds people and tells a lot about them.
- Like twitter which offers a means to convey the opinion in limited character there are no social media pages which allows you to express your mood through your favorite music.

The Limitations:

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- Even when there are many tools in the software a few limitations include the inability to convert YouTube mixes to playlist.

The Approach:

The project is based on Incremental model. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. This continues until the complete system is achieved.

The Benefits:

1. Hassle free music transfer
2. Music for all moods
3. Meeting new people with similar taste

EXP NO: 2

DATE:

2.1 Stake Holders and User Description

2.1.1 Stake Holders:

1. Developing Team:

- The team members who are developing the software are main stakeholders of the project. They have an important say and make any technical decisions related to the project. They design the final output and develop accordingly.

2. Music artists:

- The content shared and utilised are purely the work of the artists and the full copyrights related to the songs are given to them.

3. User:

- The users are the persons, who will use the project's service or result. They may be internally or externally helping the community. The users are the main concern, and the software is developed totally based on their comfort.

4. Sponsor:

- A sponsor is the person or group that provides the financial resources in kind or in cash, for the project.

5. Project Manager:

- They are assigned by the performing organization to achieve the project objective.

6. Music platforms:

- Even though the software offers many advantages, the software built uses various platforms and integrates them.

7. Portfolio Review Board:

- They are the board of committees usually made up of the organization's executive who act as a project section panel.

2.1.2 User Story

1. **Customers:**

   - A customer is an individual or business that purchases another company's goods or services. Customers are important because they drive revenues; without them, businesses have nothing to offer.

2. **IT Executive:**

   - It is completely responsible to answer the customer's queries and sort out their problems.

3. **Financial Department:**

   - This department is responsible for managing all the financial details and doing business with other organizations.

4. **Advertising Agency:**

   - The main responsibility of advertising agencies is to promote the product or services to make profit.

5. **Marketing Department:**

   - It is the Marketing Department's job to reach out to prospects, customers, investors and/or the community, while creating an overarching image that represents your company in a positive light.

6. **Data Warehouse:**

   - It is a medium for online analytical processing and data mining for improving business intelligence by turning data into information.

2.1.3 Module

Operator Module:

1. User Friendly GUI (Graphical User Interface)
2. 24/7 Customer Service
3. Generating Dynamic Playlist Link
4. Database Management
5. Data Processing
6. Software Maintenance

User Module:

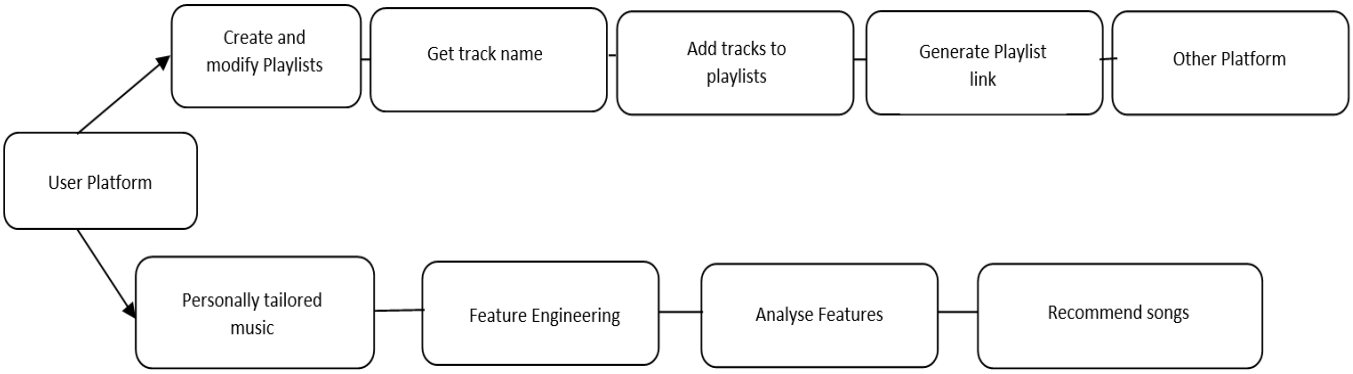
1. Modes for Input
2. Software Application for Music Mixing
3. Analyse/Compare Playlist with others
4. Get Music Recommendations
5. Feedback

2.1.4 Problem Statement

A few problems faced are

1. Unable to find exact song match
2. Server error
3. API error
4. Classic Machine Learning Errors

2.1.5 Perspective Model



2.1.6 Comparison between Waterfall and Agile Model

Why Agile Model is better than Waterfall Model?

- The Agile Model is based on iterative development and hence it divides the entire project into smaller parts which reduces the risk factor which is not the case in waterfall model.
- The Waterfall model cannot accept the changes in requirements but in agile model it is easy to change the system requirements.
- In agile model, the entire project is divided into smaller parts which helps to minimize the project risk and to reduce the overall project delivery time requirements.
- In waterfall model since risk factor is high, it is not suitable for complex projects.
- In waterfall model the testing is done in later stage it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare, which is not the case in agile model.
- In waterfall model, it follows a sequential approach whereas in agile model it explains the process in order of incremental approach.
- In agile it performs the testing concurrently with software development whereas in waterfall model the testing comes after the build phase only.
- In agile model the distance between the customer and developer is in short whereas in waterfall model it is long.
- In agile there can be done any change in the project but in waterfall model there is no changes throughout the project work

EXP NO: 3

DATE:

3.1 Identifying the Requirements from the Project Statement

3.1.1 Requirements:

Requirements are defined during the early stages of the system development as a specification of what should be implemented. A collection of requirements is a requirements document. They may be user level facility description, detailed specification of system behaviour, general system property, a specific constraint on the system or information on how-to carry-on computation. The three types of requirements are explained below.

1. System
2. Functional
3. Non-Functional

3.1.2 System Requirements:

- ✓ Spotify Developer Account
- ✓ Apple Developer Account
- ✓ Google Account
- ✓ Mid-Range PC / Laptop (Windows OS)

3.1.3 Functional Requirements:

- ✓ Link Users choice of Music Streaming Service (Spotify, Apple Music and YouTube)
- ✓ Accept input from User to Create Playlist
- ✓ Modify User's current playlist accordingly
- ✓ Generate Music to match with users' music taste (Machine Learning)

3.1.4 Non-Functional Requirements:

- ✓ Portability
- ✓ User Friendly UI and UX
- ✓ Efficiency
- ✓ Security
- ✓ Maintainability

EXP NO: 4

DATE:

4.1 Project Plan and Project Effort Based on Resources

4.1.1 Project Plan:

Project Name: Integrated Music Tools

Project Members:

1. Ashwin Ram- RA2011026020027
2. Krishnan Narayanan - RA2011026020030
3. Anuradha Arumugam - RA2011026020043

The following budget is set according to the requirements:

Resource Requirement	Cost
Computer with moderate to high level specs	₹50,000/-
Spotify Developer Account	Free
Apple Developer Account	$10/₹700/-
Python Software	Free
Misc. Charges	₹500/-
Total	**₹51,200/-**

4.1.2 Modules:

- ✓ User Authentication
- ✓ Create Playlist
- ✓ Get Playlist Information
- ✓ Modify Playlist
- ✓ Recommend Music Tracks
- ✓ Create Dynamic Playlist link to share across all Music streaming platforms

4.1.3 Scheduling:

Task	Start Date	End Date
Business Case Development	March 10th	March 17th
Identify Stake Holders and Process Models	March 17th	March 24th
Identify requirements	March 24th	March 31st
Setting Cost Estimates	March 24th	March 31st
Coding	April 1ST	June 15th
App Interface Development	June 15th	June 30th
UML Diagrams	June 30th	July 1st
Alpha Testing	July 1st	---
Final Revisions	July 1st	July 4th
Project Delivery	---	July 5st

4.1.4 Software and Hardware Requirements for the System:

- Processor: Intel i5 11th Generation
- LED Monitor
- OS: Windows 10/11
- Good Internet Connection
- Python and Music Streaming Software(s) Installed

4.2 Identify Job Roles and Responsibilities:

Member	Roles and Responsibilities
Anuradha Arumugam	Team Lead **Team Leader** has the responsibility of coordinating the team, checking in for updates and guiding the team for efficient and successful completion of the project. **Team Member** has the responsibility of contributing to the documentation. **Developer** has the responsibility of coding modules like Login and Update.
Krishnan Narayanan	Testing Lead **Testing Lead** has the responsibility to check if the actual result is matching with the expected result.

	Team Member has the responsibility of contributing to the documentation. **App Developer** has the responsibility of developing interface using Tkinter Package.
Ashwin Ram	Tech Lead **Tech Lead** has the responsibility for the overall Success of the project. **Team Member** has the responsibility of contributing to the documentation. **Developer** has the responsibility of coding other modules.

4.3 Job Role and Responsibilities:

General Definition:

In every project the group members are divided and given a certain role to accomplish the task.

Sponsor

- Project Sponsor is the individual or an organization who provides financial assistance for the development and execution of the project. They monitor the process and clarify the scope questions. They also provide expert judgment or project end date.

Subject Matter Experts

- Subject Matter Expert or SME is a person who has special skills or knowledge about a particular subject or topic. They have a deep understanding of particular process or function and hence they try to solve the technical challenges faced by the project.

Product Owner

- A Product owner is the person who is responsible for the success of the product. He/she is responsible for the backend management, stakeholder management and maximizing the value of the product.

Project Manager

- A Project Manager is a professional in the field of project management. Project Manager is responsible for planning, procurement and execution of the project.

Technical Lead

- Technical Lead is a person who is responsible for overall planning, execution and success of overall complex software solutions to meet customers' needs. They have to implement the best coding standards in the project. They have to manage the technical scope.

Software Developers

- Individuals who build and create an application are called software developers. They write, debug and then execute the source code of the software application. They are also called programmers.

Software Tester

- Software Testers are the individuals who check the actual result to match the expected result. They also ensure that software is defect free. They identify the errors and report it back to the developers' team. They either do it manually or use automated tools to detect the error.

App Developer

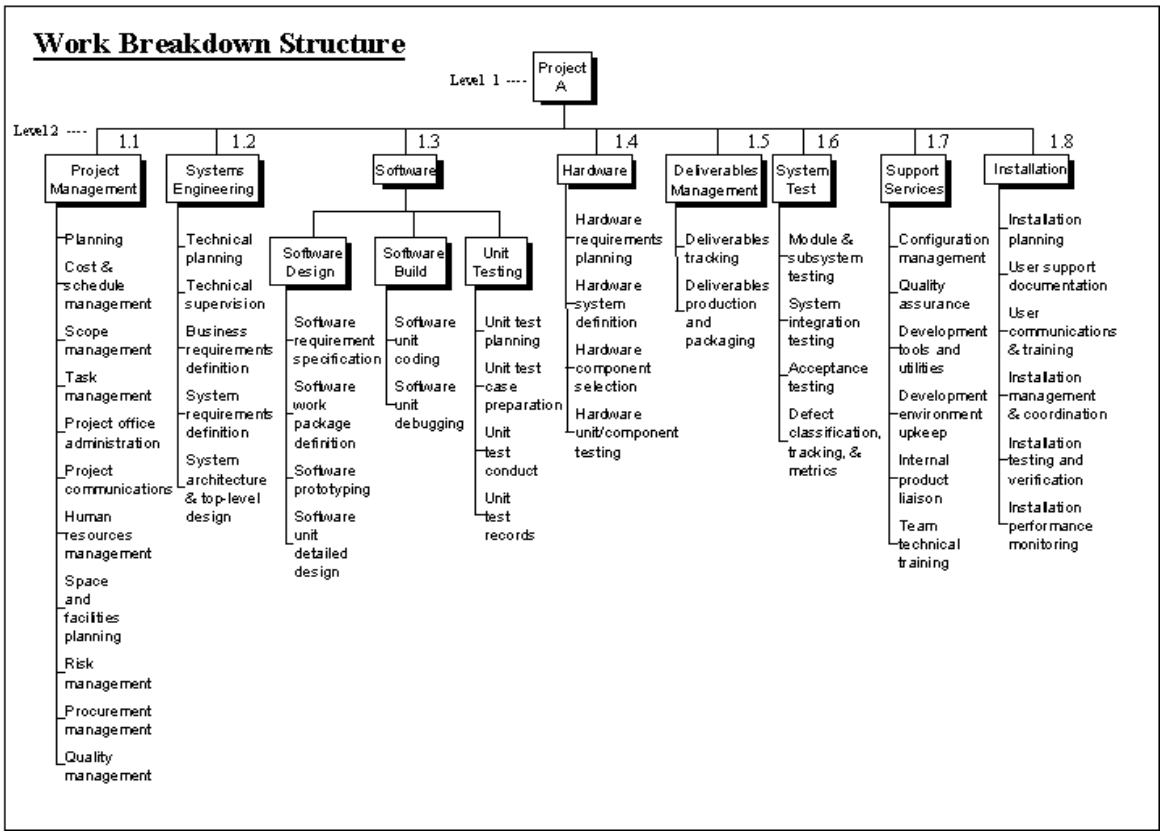
- An APP developer is a programmer who's engaged in the development of Applications using a Client server model.

EXP NO: 5

DATE:

5.1 Project Effort Based on Resource

5.1.1 Work Breakdown Structure



5.1.2 Risk Analysis

1. User Data Privacy:

- Data privacy is the protection of personal data from those who should not have access to it and the ability of individuals to determine who can access their personal information.

2. Schedule Risk:

- Schedule related risks refers to time related risks or project delivery related planning risks. The wrong schedule affects the project development and delivery. These risks are mainly indicating to running behind time as a result project development doesn't progress timely and it directly impacts to delivery of project.

3. Budget Risk:

- Budget related risks refers to the monetary risks mainly it occurs due to budget overruns. Always the financial aspect for the project should be managed as per decided but if financial aspect of project mismanaged then their budget concerns will arise by giving rise to budget risks. So proper finance distribution and management are required for the success of project otherwise it may lead to project failure.

4.Technical Risk:

- Technical risks refers to the functional risk or performance risk which means this technical risk mainly associated with functionality of product or performance part of the software product.

5. Operational Risk:

- Operational risk refers to the procedural risks means these are the risks which happen in day-to-day operational activities during project development due to improper process implementation or some external operational risks.

6. Known Risks:

- Those risks that can be uncovered after careful assessment of the project program, the business and technical environment in which the plan is being developed, and more reliable data sources (e.g., unrealistic delivery date)

7.Predictable Risks:

- Those risks that are hypothesized from previous project experience (e.g., past turnover)

8. Unpredictable risks:

- Those risks that can and do occur, but are extremely tough to identify in advance.

EXP NO: 6

DATE:

6.1 Estimation of Projects Metric's

6.1.1 Function Point Analysis

Function Point Analysis (FPA) is a method or set of rules of Functional Size Measurement. It assesses the functionality delivered to its users, based on the user's external view of the functional requirements. It measures the logical view of an application, not the physically implemented view or the internal technical view.

Types of FPA:

Transactional Functional Type –

External Input (EI): EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.

External Output (EO): EO is an elementary process that generates data or control information sent outside the application's boundary.

External Inquiries (EQ): EQ is an elementary process made up of an input-output combination that results in data retrieval.

Data Functional Type –

Internal Logical File (ILF): A user identifiable group of logically related data or control information maintained within the boundary of the application.

External Interface File (EIF): A group of users recognizable logically related data allusion to the software but maintained within the boundary of another software.

Benefits of FPA:

1. FPA is a tool to determine the size of a purchased application package by counting all the functions included in the package.
2. It is a tool to help users discover the benefit of an application package to their organization by counting functions that specifically match their requirements.
3. It is a tool to measure the units of a software product to support quality and productivity analysis.
4. It's a vehicle to estimate the cost and resources required for software development and maintenance.
5. It is a normalization factor for software comparison.

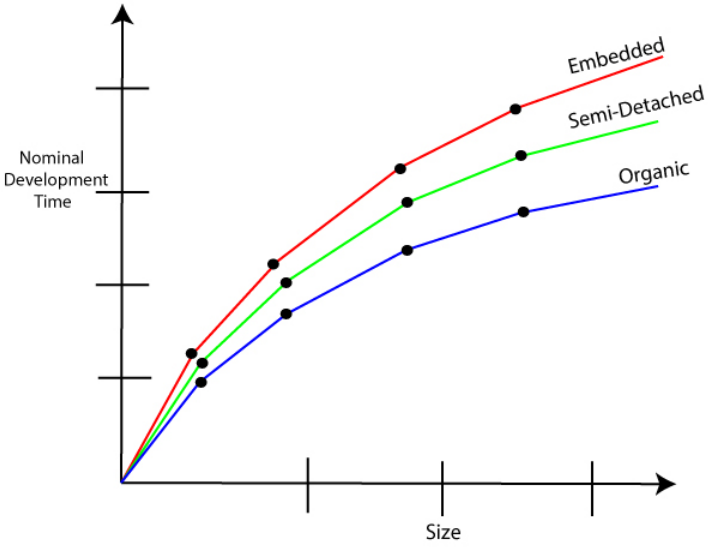
6.2.2 COCOMO Model

COCOMO (Constructive Cost Model) is a regression model based on LOC (number of Lines of Code) . It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size,

effort, cost, time, and quality. It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects, which makes it one of the best-documented models.

Types of COCOMO Models:

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements.



Development time versus size

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

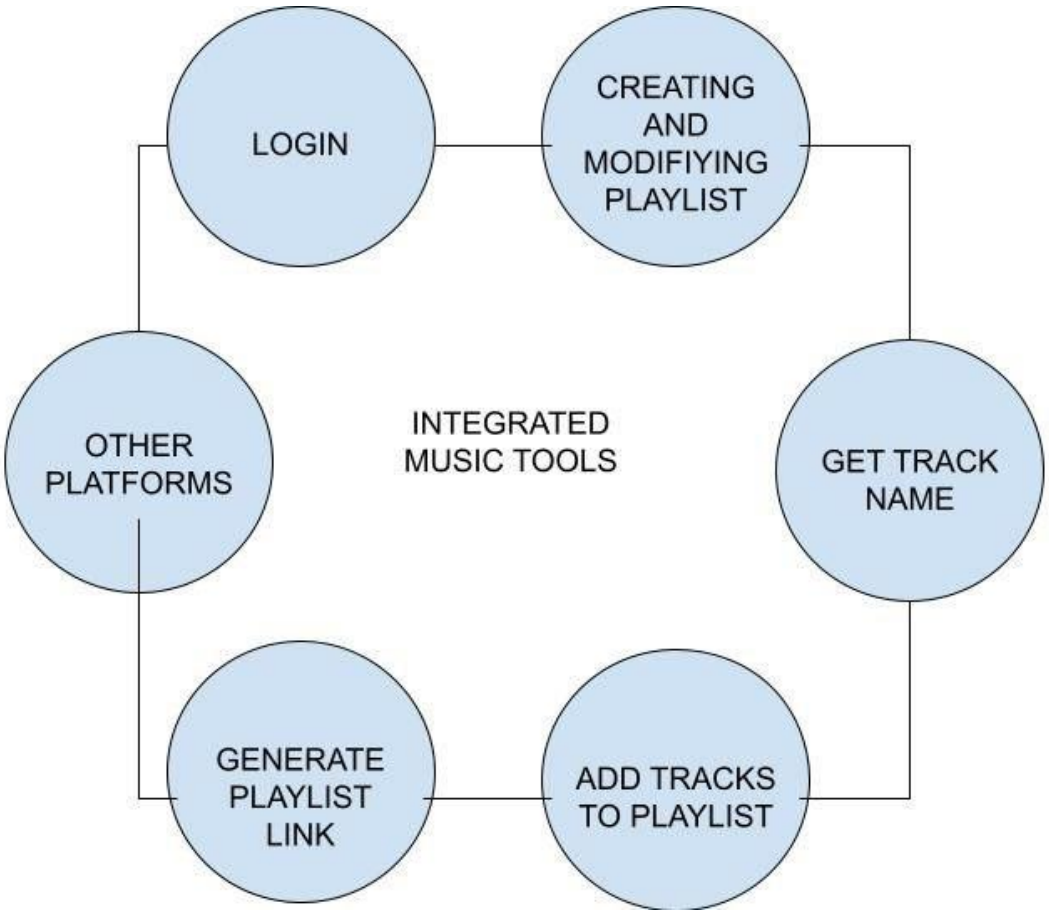
Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	..	..	1.00	1.11	1.30	1.66
STOR	..	..	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

EXP NO: 7

DATE:

7.1 DESIGN

7.1.1 SYSTEM DESIGN



The above is a simple form of system design diagram which uses front end design. This shows a loop of functions that need to be executed when this project is implemented. This is a chain of operations through which this project is implemented.

EXP NO: 8

DATE:

8.1 Modelling UML Use Case Diagrams & Capturing Use Case Scenarios

8.1.1 Use Case Diagram Description

Use Case Diagram:

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system.

Use Case Symbols and Notation:

The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams.

1. System:

A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

2. Use Cases:

Horizontally shaped ovals that represents an action which accomplishes some sort of task within the system.

3. Actors:

Stick figures that represent the people actually employing the use cases. It should be placed outside the system.

There are two types of Actors namely:

Primary Actor: Initiates the use of the system. It should be placed on the left side of the system.

Secondary Actor: It is more reactionary and should be placed on the right side of the system.

4. Relationships:

Include: This shows the dependency between base and included use case (it happens every time).

Extent: This happens only when certain criteria is met.

5. Association:

A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

The Actors used in the project are

1. User
2. Admin

The Modules used in the project are

1. User Authentication
2. Create Playlist
3. Get Playlist Information
4. Modify Playlist
5. Recommend Music Tracks
6. Generate Dynamic Playlist Link

8.1.2 Use Case Diagram